A Skew-Symmetric Form of the Recursive Newton-Euler Algorithm for the Control of Multibody Systems

Scott R. Ploen
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA., 91009-8099
E-mail: sploen@grover.jpl.nasa.gov

Abstract

In this article we derive a form of the recursive Newton-Euler algorithm that satisfies the skew-symmetry property $\dot{M} - 2C = -(\dot{M} - 2C)^T$ required in a variety of nonlinear control laws occurring throughout the fields of robotics and multibody dynamics. (Here M denotes the mass matrix of the multibody system and C denotes the Coriolis/centrifugal matrix.) We show that the recently developed formulation of multibody dynamics based on Lie groups and Lie algebras given in [11], [12] can be modified to accommodate the skew-symmetry requirement. Specifically, we demonstrate that explicit block-triangular factorizations of both M and C are embedded within the structure of the recursive algorithm. Furthermore, the factorization of the mass matrix M can be differentiated explicitly with respect to time. The resulting expressions for $M, \dot{M}$, and C immediately lead to a proof based entirely on high-level matrix manipulations demonstrating the skew-symmetry of $\dot{M} - 2C$.

1 Introduction

As demonstrated in any standard text on analytical dynamics (e.g., [3], [8]), the equation of motion of a large class of mechanical systems (i.e., *natural systems* - defined as systems whose kinetic energy is of the form $\frac{1}{2}\dot{q}^T M(q)\dot{q}$) can be expressed as follows

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \phi(q) = \tau \tag{1}$$

where $q = (q_1, q_2, \cdots, q_n) \in \Re^n$ denotes a vector of generalized coordinates describing the time evolution of the system, $M(q) \in \Re^{n\times n}$ is the symmetric positive definite mass matrix, $C(q,\dot{q}) \in \Re^{n\times n}$ is the Coriolis-centrifugal matrix, $\phi(q) \in \Re^n$ is the vector of gravity terms, and $\tau \in \Re^n$ denotes the applied loading. It is well known (e.g., see

[10],[14]) that although the matrix-vector product $C\dot{q}$ is unique for a given multibody system, the matrix C is not. This is a consequence of the fact that the Coriolis matrix C is *itself a function of* $\dot{q}$.

For the purpose of dynamic simulation (i.e., determining the motion of the system via numerical integration when the applied loading is specified) the non-uniqueness of the Coriolis/centrifugal matrix is not an important issue as any admissible C matrix leads to same acceleration vector $\ddot{q}$. (In fact, there is no reason at all to express the Coriolis vector as a matrix-vector product for dynamic simulation.) However, in a variety of nonlinear control problems in robotics and multibody dynamics the choice of the Coriolis/centrifugal matrix C is critical. For example, the skew-symmetry of $\dot{M}-2C$ plays an important role in developing Lyapunov functions for control laws used in the control of large scale flexible space structures [7]. As another example, Slotine and Li [14] have developed a globally stable adaptive control law for robotic systems that results in asymptotic tracking of a desired reference trajectory $q_d(t) \in \Re^n$. The stability proof of the control law requires that $s^T(\dot{M} - 2C)s = 0$ where $s \in \Re^n$ is a function of both q and $\dot{q}$. (See [14] for more details.) As a result, C must be constructed in such a way to render $(\dot{M} - 2C)$ skew-symmetric. As an aside, we note an important fact which has caused some confusion in the past, viz., if $s = \dot{q}$ then $\dot{q}^T(\dot{M} - 2C)\dot{q} = 0$ irrespective of the skew-symmetry of $\dot{M} - 2C$. This statement is a property of finite-dimensional natural systems and its proof can be found in [10] and [14]. As a result, it is only in situations where $(\dot{M} - 2C)$ is pre and post multiplied by a vector different from $\dot{q}$ that C must be carefully defined.

It is well known ([10], [14]) that if C is defined as

$$C_{kj} = \sum_{i=1}^{n} \Gamma_{ijk}\dot{q}_i \tag{2}$$

where Γ_{ijk} denote the *Christoffel symbols of the first kind*

$$\Gamma_{ijk} = \frac{1}{2}(\frac{\partial M_{kj}}{\partial q_i} + \frac{\partial M_{ki}}{\partial q_j} - \frac{\partial M_{ij}}{\partial q_k}) \tag{3}$$

and M_{ij} are the elements of the mass matrix, then $\dot{M} - 2C$ will be skew-symmetric. A major drawback of using equation (2) to construct C is that the entries of the mass matrix *must be known a priori* and *in an explicit fashion* in order to compute the partial derivatives. Anyone who has derived the mass matrix for even a simple multibody system however, will have experienced firsthand the enormous complexity of the resulting equations. (See [1] for an example of the symbolic complexity of the mass matrix of a PUMA 560 robot - a system with relatively simple open chain topology!) A second drawback associated with the use of equation (2) is that this particular definition of C is intimately related to a Lagrangian formulation of the equations of motion [3]. For many applications in multibody dynamics and control the dynamic equations are expressed recursively (usually in terms of Newton's second law and Euler's equations) and the above definition of C does not readily apply.

The remainder of the article is organized as follows. After a brief review of Lie theory, we demonstrate that the recursive formulation of multibody dynamics based on Lie groups given in [11] can be recast into global matrix form via a series of simple linear algebraic operations. In the resulting set of closed-form equations, the mass matrix and Coriolis/centrigual matrix admit concise block-triangular factorizations in which the kinematic and dynamic parameters of the robot appear transparently. Further, the factorization of the mass matrix can be differentiated explicitly with respect to time. These high-level matrix expressions for $M, \dot{M}$ and C immediately lead to a simple matrix expression for $\dot{M}-2C$. Unfortunately, the Coriolis/centrifugal matrix C associated with the recursive algorithm in its original form does not render $\dot{M}-2C$ skew-symmetric. Next, we demonstrate that the equations of motion of the entire system inherit the skew-symmetry property from the equations of motion of each individual body. (This idea was developed independently by Lin *et al* [5] although we present a far more transparent derivation.) Using this idea we show how to construct a skew-symmetric form of the recursive Newton-Euler algorithm. The modified recursive algorithm is then recast into global matrix form and a new matrix factorization of the Coriolis/centrifugal matrix C is determined. We then present a proof demonstrating the skew-symmetry of $\dot{M}-2C$ based entirely on simple matrix manipulations. This is in contrast to the proof given by Lin *et al* in [5] which involves index notation and multiple summations.

2 Mathematical Background

2.1 SE(3), se(3), and se(3)*

In this section we give a brief review of the material from Lie theory needed in the sequel. The reader is referred to [9], [12], [4] for a more comprehensive discussion.

Given an inertially fixed reference frame F, the position and orientation of a rigid body $X = (\Theta, b)$ is described by an element of the *Special Euclidean Group* of rigid-body motions, denoted $SE(3)$, consisting of matrices of the form

$$\begin{bmatrix} \Theta & b \\ 0 & 1 \end{bmatrix}$$

where $\Theta \in SO(3)$ and $b \in \Re^3$. Here $SO(3)$ denotes the group of 3×3 proper orthogonal matrices. SE(3) has the structure of both a mathematical group under matrix multiplication and a differentiable manifold and is therefore a (matrix) *Lie Group.* The generalized velocity of a rigid body $x = (\omega, v)$ is described by an element of the *Lie algebra* of $SE(3)$, denoted $se(3)$, consisting of matrices of the form

$$\begin{bmatrix} [\omega] & v \\ 0 & 0 \end{bmatrix}$$

where

$$[\omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

and $v \in \Re^3$. The Lie algebra se(3) is the tangent space of SE(3) at the identity element of the group.

Generalized forces acting on a rigid body $F = (m, f)$ are described by elements of the dual space to $se(3)$, denoted $se(3)^*$. The dual space $se(3)^*$ is the space of *linear functionals* on $se(3)$: i.e., if $V = (\omega, v) \in se(3)$ and $F = (m, f) \in se(3)^*$ then $F : se(3) \mapsto \Re$ is given by $F(V) = F^T V = m^T w + f^T v$. Note that $F(V)$ has units of power.

An element of a Lie group can also be identified with a linear mapping between its Lie algebra via the *adjoint representation.* If $X = (\Theta, b)$ is an element of $SE(3)$, then its adjoint map $Ad_X : se(3) \mapsto se(3)$ admits the 6×6 matrix representation

$$Ad_X(x) = \begin{bmatrix} \Theta & 0 \\ [b]\,\Theta & \Theta \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix} \tag{4}$$

where $[b]$ denotes the 3×3 skew-symmetric matrix representation of $b \in \Re^3$. Physically, the adjoint mapping describes how generalized velocities transform under a change of reference frame given by $X = (\Theta, b)$. For example, if $V_2 = (w_2, v_2)$ denotes the generalized velocity of a rigid body with respect to a reference frame M_2, and $V_1 = (w_1, v_1)$ denotes the generalized velocity of the same body with respect to a reference frame M_1, and $X_{12} = (\Theta_{12}, b_{12})$ represent the position and orientation of M_2 relative to M_1, then $V_1 = Ad_{X_{12}} V_2$. It is easily verified that $Ad_X^{-1} = Ad_{X^{-1}}$ and $Ad_X Ad_Y = Ad_{XY}$ for any $X, Y \in SE(3)$.

The dual operator $Ad_X^* : se(3)^* \mapsto se(3)^*$ admits the matrix representation given by the transpose of Ad_X; i.e., if $z = (m, f) \in se(3)^*$, then

$$Ad_X^*(z) = \begin{bmatrix} \Theta^T & \Theta^T [b]^T \\ 0 & \Theta^T \end{bmatrix} \begin{bmatrix} m \\ f \end{bmatrix}. \tag{5}$$

Physically, the dual adjoint mapping describes how generalized forces transform under a change of reference frame given by $X = (\Theta, b)$. For example, if $F_2 = (m_2, f_2)$ denotes the generalized force acting on a rigid body with respect to a reference frame M_2, and $F_1 = (m_1, f_1)$ denotes the generalized force acting on the same body with respect to a reference frame M_1, and $X_{12} = (\Theta_{12}, b_{12})$ represents the position and orientation of M_2 relative to M_1, then $F_2 = Ad_{X_{12}}^* F_1$.

Elements of the Lie algebra can also be identified with a linear mapping between the Lie algebra and itself via the *Lie bracket.* On matrix Lie algebras the Lie bracket is given by the matrix commutator: viz., if $A, B \in se(3)$, then $[A, B] = AB - BA \in se(3)$. Given $x \in se(3)$ its adjoint representation is given by the linear map defined by $ad_x(y) = [x, y]$. For $x = (\omega_1, v_1)$ and $y = (\omega_2, v_2) \in se(3)$, the adjoint map admits the following 6×6 matrix representation:

$$ad_x(y) = \begin{bmatrix} [\omega_1] & 0 \\ [v_1] & [\omega_1] \end{bmatrix} \begin{bmatrix} \omega_2 \\ v_2 \end{bmatrix}. \tag{6}$$

Similarly, the matrix representation of the dual operator $ad_x^* : se(3)^* \mapsto se(3)^*$ is the matrix transpose of ad_x:

$$ad_x^*(z) = \begin{bmatrix} -[\omega_1] & -[v_1] \\ 0 & -[\omega_1] \end{bmatrix} \begin{bmatrix} m \\ f \end{bmatrix}. \tag{7}$$

Physically, the mappings $ad_x(y)$ and $ad_x^*(z)$ can be thought of as generalizations of the standard cross product operation to $se(3)$ and $se(3)^*$ respectively.

2.2 The Product of Exponentials Formula

We now briefly review the product of exponentials (POE) formula for the kinematic analysis of of multibody systems. For a more detailed discussion see [2] or [9]. If a dextral reference frame is fixed on each body in a multibody system where each body is connected via single degree of freedom joints, then the element of $SE(3)$ describing the position and orientation of frame i relative to frame $i-1$ is given by$f_{i-1,i} = M_i e^{P_i q_i}$, where $M_i \in SE(3)$, $P_i \in se(3)$, and $q_i \in \Re$ is the joint variable for link i. Physically $f_{i-1,i}$ represents the coordinate transformation across a rigid body in a multibody system. The frame fixed at the tip of the kinematic chain is then related to that of the base by the product

$$f(q_1, \ldots, q_n) = M_1 e^{P_1 q_1} \ldots M_n e^{P_n q_n}. \tag{8}$$

The matrix exponentials in the above formula can be computed in closed-form via the following result: Let $(\omega, v) \in se(3)$ where ω has unit length. Then for any $\phi \in \Re$,

$$\exp\left(\begin{bmatrix} [\omega] & v \\ 0 & 0 \end{bmatrix} \phi \right) = \begin{bmatrix} \exp([\omega]\,\phi) & b \\ 0 & 1 \end{bmatrix} \tag{9}$$

is an element of $SE(3)$, where

$$\exp([\omega]\,\phi) = I + \sin\phi\,[\omega] + (1-\cos\phi)[\omega]^2 \tag{10}$$

$$b = \left(\phi I + (1-\cos\phi)\,[\omega] + (\phi - \sin\phi)[\omega]^2\right) v \tag{11}$$

3 The Equations of Motion

3.1 Recursive Newton-Euler Algorithm

We now review the Lie group based recursive formulation of dynamics as given in [11] and [12]. The idea behind the recursive formulation is a two-stage iterative process. In the outward iterative stage the generalized velocities and accelerations of each body are propagated from the base to the tip, each quantity expressed in local body fixed coordinates. In the inward iterative stage the generalized forces are propagated backward from the tip body to the base body, also expressed with respect to local body frame coordinates. We make the following definitions (again, all quantities are expressed in the corresponding local body fixed coordinates): let $V_i \in \Re^{6\times 1}$ denote the generalized velocity of body i, $F_i \in \Re^{6\times 1}$ the total generalized force transmitted from body $i-1$ to body i through joint i with its first three components corresponding to the moment vector, and τ_i the applied torque/force at joint i. Also, let $f_{i-1,i} = M_i e^{S_i q_i}$ denote the position and orientation of the body i frame relative to the body $i-1$

frame with $M_i \in SE(3)$ and $S_i = (\omega_i, 0) \in se(3)$ (here ω_i is a unit vector along the axis of rotation of joint i). Further $J_i \in \Re^{6\times 6}$ is defined as

$$J_i = \begin{bmatrix} I_i - m_i[r_i]^2 & m_i\,[r_i] \\ -m_i\,[r_i] & m_i \cdot \mathbf{1} \end{bmatrix} \tag{12}$$

where m_i is the mass of body i, r_i is the vector in body i coordinates from the origin of the body i frame to the center of mass of body i, and I_i is the inertia tensor of body i about the center of mass. Here $\mathbf{1}$ denotes the 3×3 identity matrix.

The recursive Newton-Euler algorithm can now be expressed in terms of our earlier geometric definitions in the following manner:

- **Initialization**

$$Given: \quad V_0, \dot{V}_0, F_{n+1} \tag{13}$$

- **Forward recursion: for i = 1 to n do**

$$f_{i-1,i} = M_i e^{S_i q_i} \tag{14}$$

$$V_i = Ad_{f^{-1}_{i-1,i}}(V_{i-1}) + S_i \dot{q}_i \tag{15}$$

$$\begin{aligned} \dot{V}_i &= S_i \ddot{q}_i + Ad_{f^{-1}_{i-1,i}}(\dot{V}_{i-1}) \\ &\quad - ad_{S_i \dot{q}_i} Ad_{f^{-1}_{i-1,i}}(V_{i-1}) \end{aligned} \tag{16}$$

- **Backward recursion: for i = n to 1 do**

$$\begin{aligned} F_i &= Ad^*_{f^{-1}_{i,i+1}}(F_{i+1}) + J_i \dot{V}_i \\ &\quad - ad^*_{V_i}(J_i V_i) \end{aligned} \tag{17}$$

$$\tau_i = S_i^T F_i \tag{18}$$

Here V_0 and $\dot{V}_0$ denote the generalized velocity and acceleration of the base respectively, and F_{n+1} (assumed to be zero in the sequel) denotes the generalized force acting at the tip of the multibody chain. The recursive algorithm presented above is valid for open chain multibody systems consisting of bodies connected via single degree-of-freedom joints (e.g., revolute or prismatic joints). These assumptions can be relaxed and the above algorithm can be extended to multibody systems with arbitrary tree-topology structure and multi-degree-of-freedom joints. (See [12] for further information.) Although not discussed here, the above recursive algorithm is also completely independent of the reference frames chosen to express the equations of motion. See [12] for more details.

3.2 Global Matrix Representation of the Newton-Euler Algorithm

By expanding the individual equations (15)-(18) and rearranging it can be shown that the recursive Newton-Euler algorithm admits the following global matrix representation:

$$V = GS\dot{q} + GP_0V_0 \tag{19}$$

$$
\begin{aligned}
\dot{V} &= GS\ddot{q} + Gad_{S\dot{q}}\Gamma V + Gad_{S\dot{q}}P_0 V_0 + GP_0\dot{V}_0 && (20)\\
F &= G^T J\dot{V} + G^T ad_V^* JV && (21)\\
\tau &= S^T F && (22)
\end{aligned}
$$

where

$$
\begin{aligned}
V &= \text{column}[V_1, V_2, \cdots, V_n] \in \Re^{6n\times 1}\\
F &= \text{column}[F_1, F_2, \cdots, F_n] \in \Re^{6n\times 1}\\
\dot{q} &= \text{column}[\dot{q}_1, \dot{q}_2, \cdots, \dot{q}_n] \in \Re^{n\times 1}\\
\tau &= \text{column}[\tau_1, \tau_2, \cdots, \tau_n] \in \Re^{n\times 1}\\
P_0 &= \text{column}[Ad_{f_{0,1}^{-1}}, 0, \cdots, 0] \in \Re^{6n\times 6}\\
S &= \text{diag}[S_1, S_2, \cdots, S_n] \in \Re^{6n\times n}\\
J &= \text{diag}[J_1, J_2, \cdots, J_n] \in \Re^{6n\times 6n}\\
ad_{S\dot{q}} &= \text{diag}[-ad_{S_1\dot{q}_1}, \cdots, -ad_{S_n\dot{q}_n}] \in \Re^{6n\times 6n}\\
ad_V^* &= \text{diag}[-ad_{V_1}^*, \cdots, -ad_{V_n}^*] \in \Re^{6n\times 6n}
\end{aligned}
$$

Here $\Gamma \in \Re^{6n\times 6n}$ is given by

$$
\Gamma = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0\\
Ad_{f_{1,2}^{-1}} & 0 & \cdots & 0 & 0\\
0 & Ad_{f_{2,3}^{-1}} & \cdots & 0 & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots\\
0 & 0 & \cdots & Ad_{f_{n-1,n}^{-1}} & 0
\end{bmatrix}
$$

Note that the eigenvalues of Γ are identically zero. As a result Γ is a nilpotent matrix, viz. $\Gamma^n = 0$, and it can easily be shown that $G = (I - \Gamma)^{-1} = I + \Gamma + \cdots + \Gamma^{n-1}$. As a result,

$$
G = \begin{bmatrix}
I_{6\times 6} & 0 & 0 & \cdots & 0\\
Ad_{f_{1,2}^{-1}} & I_{6\times 6} & 0 & \cdots & 0\\
Ad_{f_{1,3}^{-1}} & Ad_{f_{2,3}^{-1}} & I_{6\times 6} & \cdots & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
Ad_{f_{1,n}^{-1}} & Ad_{f_{2,n}^{-1}} & Ad_{f_{3,n}^{-1}} & \cdots & I_{6\times 6}
\end{bmatrix}.
$$

where $I_{6\times 6}$ denotes the 6×6 identity matrix.

In the sequel we will assume that $V_0 = 0$ and that $\dot{V}_0 = (0, g)$ where $g \in \Re^3$ denotes the gravity vector in appropriate units and direction.

Combining (19)-(22), the equations of motion can be expressed as

$$
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + \phi(q) = \tau, \qquad (23)
$$

where

$$
\begin{aligned}
M(q) &= S^T G^T JGS && (24)\\
C(q, \dot{q}) &= S^T G^T (JG\, ad_{S\dot{q}}\Gamma + ad_V^* J)GS && (25)\\
\phi(q) &= S^T G^T JGP_0\dot{V}_0 && (26)
\end{aligned}
$$

The above matrix factorization of the equations of motion is an explicit matrix representation of the $O(n)$ recursive dynamics algorithm. One of the most useful structural features of the above equations is the transparent manner in which the parameters of the multibody system appear; for example, in the factorization of the mass matrix all the inertial parameters are contained in the constant block-diagonal matrix J, while S is a constant matrix containing only the kinematic parameters, and G is the only matrix dependent on the q_i.

4 Differentiation of the System Mass Matrix

For many applications in dynamics and control the ability to differentiate the equations of motion at a high-level is of paramount importance. In this section we demonstrate that the equations of motion resulting from our Lie group formulation of multibody dynamics can be differentiated in a straightforward manner: This property is a consequence of the fact that the basic mathematical primitive on which the entire geometric formulation is based is the matrix exponential, and it is well known from linear systems theory that the derivative of e^{At} with respect to t is simply Ae^{At}. (The ability to differentiate the equations of motion explicitly is not unique to our geometric formulation of dynamics. As shown in [6] the spatial operators of Rodriguez *et al* [13] can also be differentiated at a high level. However, differentiation of the spatial operators is less transparent because the spatial operator framework is not formulated in terms of matrix exponentials.)

It follows from the expression for M given in (24) that $\frac{d}{dt}M = \frac{d}{dt}(S^T G^T J G S)$. Recalling that both S and J are constant matrices it immediately follows that $\dot{M} = S^T \dot{G}^T J G S + S^T G^T J \dot{G} S$. From comparing (19) and (20) we find

$$\dot{G} = G ad_{S\dot{q}} \Gamma G \tag{27}$$

and therefore

$$\dot{M} = S^T G^T \Gamma^T ad_{S\dot{q}}^T G^T J G S + S^T G^T J G ad_{S\dot{q}} \Gamma G S \tag{28}$$

5 Skew-Symmetry of the Newton-Euler Algorithm

In this section we address the main topic of this article. Although an explicit formula for the Coriolis/centrifugal matrix is given in (25) this particular realization of C does not render $\dot{M} - 2C$ skew-symmetric. To see this, we use (25) and (28) and find

$$\dot{M} - 2C = N - N^T - 2 S^T G^T ad_V^* J G S \tag{29}$$

where $N = S^T G^T \Gamma^T ad_{S\dot{q}}^T G^T J G S$. As $N - N^T$ is clearly skew-symmetric, the skew-symmetry of $\dot{M} - 2C$ depends on the skew-symmetry of the operator $ad_V^* J \in \Re^{6n \times 6n}$. A straightforward calculation using the definitions of $ad_V^* J$ and J given in section 3 along with (7) and (12) demonstrates that this operator *is not* skew-symmetric because the matrices $-ad_{V_i}^* J_i$ are not skew-symmetric.

Using our Lie group formulation it is clear that C fails to be skew-symmetric because the individual Newton-Euler equations for each body do not themselves satisfy the skew-symmetry property $(\dot{J}_i - 2C_i)^T = -(\dot{J}_i - 2C_i)$. To see this, recall from (17) that the Newton-Euler equations for the i^{th} body in the multibody system are

$$F_i - Ad^*_{f^{-1}_{i,i+1}}(F_{i+1}) = J_i\dot{V}_i + C_iV_i \tag{30}$$

where $C_i = -ad^*_{V_i}J_i \in \Re^{6\times 6}$. Since J_i is a constant matrix the skew-symmetry requirement for the individual Newton-Euler equations for each body reduces to $C_i^T = -C_i$. As discussed above $C_i = -ad^*_{V_i}J_i$ is not a skew symmetric matrix, and as a result $ad^*_V J = \mathrm{diag}[C_1, \cdots, C_n] \in \Re^{6n\times 6n}$ also fails to be skew-symmetric. Therefore, it is clear that the global matrix factorization of the recursive algorithm inherits the skew-symmetry property from the equations of motion on the individual body level. As a result, in order to satisfy the skew-symmetry requirement a modified Coriolis/centrifugal matrix in (30), say $\bar{C}_i$, satisfying $\bar{C}_i^T = -\bar{C}_i$ must be constructed. To this end, we present the following result:

Proposition 1

$$C_iV_i = \bar{C}_iV_i$$

where $C_i = -ad^*_{V_i}J_i$ *and the skew-symmetric matrix* $\bar{C}_i$ *is given by*

$$\bar{C}_i = \begin{bmatrix} [\omega_i]\,\bar{I}_i + \bar{I}_i\,[\omega_i] & m_i\,[r_i]\,[\omega_i] \\ -m_i\,[\omega_i]\,[r_i] & m_i\,[\omega_i] \end{bmatrix}$$

Here $\bar{I}_i = I_i - m_i[r_i]^2$.

Proof: It follows from (7) and (12) that

$$C_iV_i = \begin{bmatrix} [\omega_i] & [v_i] \\ 0 & [\omega_i] \end{bmatrix} \begin{bmatrix} I_i - m_i[r_i]^2 & m_i\,[r_i] \\ -m_i\,[r_i] & m_i\cdot\mathbf{1} \end{bmatrix} \begin{bmatrix} \omega_i \\ v_i \end{bmatrix} \tag{31}$$

Expanding out (31) and using the definition of $\bar{I}_i$ results in

$$C_iV_i = \begin{bmatrix} [\omega_i]\,\bar{I}_i\omega_i + m_i\,[\omega_i]\,[r_i]\,v_i - m_i\,[v_i]\,[r_i]\,\omega_i + m_i\,[v_i]\,v_i \\ -m_i\,[\omega_i]\,[r_i]\,\omega_i + m_i\,[\omega_i]\,v_i \end{bmatrix} \tag{32}$$

Recalling that $[v_i]\,v_i = v_i \times v_i = 0$, $-m_i\,[v_i]\,[r_i]\,\omega_i = m_i\,[v_i]\,[\omega_i]\,r_i$ and applying the Jacobi identity $[a]\,[b]\,c + [b]\,[c]\,a = -\,[c]\,[a]\,b$ to $m_i\,[\omega_i]\,[r_i]\,v_i + m_i\,[v_i]\,[\omega_i]\,r_i$ yields

$$C_iV_i = \begin{bmatrix} [\omega_i]\,\bar{I}_i\omega_i + m_i\,[r_i]\,[\omega_i]\,v_i + \bar{I}_i\,[\omega_i]\,\omega_i \\ -m_i\,[\omega_i]\,[r_i]\,\omega_i + m_i\,[\omega_i]\,v_i \end{bmatrix} \tag{33}$$

Note that we have added zero to the above expression by appending $\bar{I}_i\,[\omega_i]\,\omega_i = 0$. Rearranging (33) it immediately follows that

$$C_iV_i = \begin{bmatrix} [\omega_i]\,\bar{I}_i + \bar{I}_i\,[\omega_i] & m_i\,[r_i]\,[\omega_i] \\ -m_i\,[\omega_i]\,[r_i] & m_i\,[\omega_i] \end{bmatrix} \begin{bmatrix} \omega_i \\ v_i \end{bmatrix} \tag{34}$$

which proves the result. □

Although Lin *et al* [5] independently demonstrated that the skew-symmetry of $\dot{M} - 2C$ is inherited from the skew-symmetry of $\dot{J}_i - 2C_i$, the derivation presented above is a more explicit statement of this result.

Using the definition of $\bar{C}_i$ given in Proposition 1 the recursive algorithm (13)-(18) can be made to satisfy the skew-symmetry requirement by replacing $C_i = -ad^*_{V_i} J_i$ in (17) with $\bar{C}_i$ as given in Proposition 2. Next, upon repeating the steps leading from the recursive algorithm (13)-(18) to (23), it can be shown that the equations of motion are

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \phi(q) = \tau \tag{35}$$

where

$$\begin{aligned} M(q) &= S^T G^T J G S & (36)\\ C(q,\dot{q}) &= S^T G^T (J G ad_{S\dot{q}} \Gamma + \bar{C}) G S & (37)\\ \phi(q) &= S^T G^T J G P_0 \dot{V}_0 & (38) \end{aligned}$$

and $\bar{C} = \text{diag}[\bar{C}_1, \bar{C}_2, \cdots, \bar{C}_n] \in \Re^{6n\times 6n}$ where the $\bar{C}_i$ are as given in Proposition 1. Note that $\bar{C}$ is a skew-symmetric matrix by construction.

Proposition 2 *If C is defined as in (37) then $\dot{M} - 2C = -(\dot{M} - 2C)^T$*

Proof: Recalling from (28) that $\dot{M} = S^T G^T \Gamma^T ad^T_{S\dot{q}} G^T J G S + S^T G^T J G ad_{S\dot{q}} \Gamma G S$ it is straightforward to show that $\dot{M} - 2C = N - N^T - 2S^T G^T \bar{C} G S$ where $N = S^T G^T \Gamma^T ad^T_{S\dot{q}} G^T J G S$. Therefore $(\dot{M} - 2C)^T = N^T - N - 2S^T G^T \bar{C}^T G S$. Since $\bar{C}^T = -\bar{C}$, the result follows. □

The proof given in Proposition 2 demonstrating the skew-symmetry of $\dot{M} - 2C$ is based entirely on high-level matrix manipulations: This is in contrast to the proof given in [5] which is based on a less direct approach involving index notation and multiple summations.

6 Conclusion

In this article we have presented a version of our Lie group based multibody dynamics algorithm which satisfies the skew-symmetry property required in a variety of control applications in robotics and structural dynamics. We have shown that the Lie group formulation of dynamics leads to explicit factorizations of both the mass and Coriolis matrices. Furthermore, we have demonstrated that the mass matrix can be differentiated explicitly with respect to time. The resulting expressions for M,$\dot{M}$, and C immediately lead to a proof of the skew-symmetry of $\dot{M} - 2C$ based entirely on high-level matrix manipulations.

Acknowledgments

The research described in this article was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

[1] B. Armstrong, O. Khatib, and J. Burdick, "The explicit model and inertial parameters of the PUMA 560 robot arm," *Proc. IEEE Int. Conf. Robotics and Automation.* San Francisco, CA: IEEE, pp. 510-518, 1986.

[2] R.W. Brockett, "Robotic manipulators and the product of exponentials formula," *Proc. Int. Symp. Math. Theory of Networks and Systems,* Beer Sheba, Israel, pp. 120-129, 1983.

[3] D.T. Greenwood, *Classical Dynamics.* Englewood Cliffs, NJ: Prentice Hall, 1977.

[4] R. Gilmore, *Lie Groups, Lie Algebras, and Some of Their Applications.* New York, John Wiley, 1974.

[5] H.C. Lin, T.C. Lin, and K.H. Yae, "On the skew-symmetric property of the Newton-Euler formulation for open chain robot manipulators," *Proc. American Control Conf.* Seattle, WA, pp. 2322–2326, 1995.

[6] A. Jain, and G. Rodriguez, "Diagonalized Lagrangian robot dynamics," *IEEE Trans. Robotics and Automation.* Vol. 11, No. 4, pp.571-584, 1995.

[7] A. Kelkar, and S. Joshi, *Control of Nonlinear Multibody Flexible Space Structures: Lecture Notes in Control and Information Sciences, Volume 221.* New York, Springer, 1996

[8] L. Meirovitch, *Methods of Analytical Dynamics.* New York: McGraw-Hill, 1970.

[9] R.M. Murray, Z. Li, and S.S. Sastry, *A Mathematical Introduction to Robot Manipulation.* Boca Raton, FL: CRC Press, 1993.

[10] R. Ortega, and M.W. Spong, "Adaptive motion control of rigid robots: a tutorial," *Proc. IEEE Conf. on Decision and Control.* Austin, TX: IEEE, pp. 1575-1584, 1988.

[11] F.C. Park, J.E. Bobrow, and S.R. Ploen, "A Lie group formulation of robot dynamics," *Int. J. Robotics Research.* Vol. 14, No. 6, pp. 609-618, 1995.

[12] S.R. Ploen, *Geometric Algorithms for the Dynamics and Control of Multibody Systems.* Ph.D. Thesis, University of California, Irvine, 1997.

[13] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "A spatial operator algebra for manipulator modeling and control," *Int. J. Robotics Research.* Vol. 10, No. 4, pp. 371-381, 1991.

[14] J.J.E. Slotine, and W. Li, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice Hall, 1991.